In the claims:

All of the claims standing for examination are reproduced below. Claims 1 and 13 are amended in this response.

1. (Currently amended) A system architecture for ~~adapting at least two legacy systems for functional interface with at least one component system comprising~~ providing interaction between legacy and component systems, comprising:

two or more legacy systems:

a data reconciliation bus for data redundancy between at least two of the two or more legacy systems;

at least one component wrapper within the architecture for describing the two or more legacy systems;

at least one component object within the architecture for describing the at least one component system; and

a connectivity bus within the architecture between the at least one component object and the at least one component wrapper, for extending legacy function to the at least one component system;

characterized in that a user operating a GUI has access to legacy services in an automated client/server exchange wherein heterogeneous data formats and platform differences of the separate systems are resolved in an object-oriented way that is transparent to the user, and further characterized in that the data reconciliation bus utilizes an in-memory entity-relationship (ER) model of each legacy system of the system architecture.

2. (Previously presented) The system architecture of claim 1 wherein one component object is interfaced with more than one legacy system in the event of more than one

system.

3. (Original) The system architecture of claim 1 wherein one legacy system is interfaced with more than one component system in the event of more than one component system.

4. (Canceled).

5. (Original) The system architecture of claim 1 wherein entity-relationship modeling is used to model legacy services.

6. (Original) The system architecture of claim 5 wherein a component wrapper is completely generated from an object model of legacy services.

7. (Previously presented) The system architecture of claim 1 wherein heterogeneity in data formats between a legacy system and a component wrapper is resolved by a language adapter interface.

8-12 (Canceled)

13. (Currently amended) In a system architecture for integrating legacy systems and component systems, a data reconciliation framework for achieving data reconciliation between redundant data elements in the legacy systems comprising:

a memory component with a data model stored therein, the data model describing all legacy systems data and component systems data;

a first function for propagating data from a two or more legacy ~~system~~ systems; and

a second function for propagating data to ~~a legacy system or~~ the two or more

legacy systems;

characterized in that the first function updates the data model and the second function takes the update from the data model as input and propagates it to the appropriate system or systems, and further characterized in that the data reconciliation framework utilizes an in-memory entity-relationship (ER) model of each legacy system of the system architecture.

14. (Original) The data reconciliation framework of claim 13 wherein the data model stored in memory is a unified normalized layer.

15. (Original) The data reconciliation framework of claim 13 wherein the first and second functions are automated.

16. (Original) The data reconciliation framework of claim 13 wherein the first and second functions are user executed.

17. (Original) The data reconciliation framework of claim 13 wherein the functions propagate data in an object oriented environment.